FRANCESCO ANTICI

SAT CONSTRAINTS EMBEDDINGS

CARDINALITY CONSTRAINTS

- In most of the exercises we solved we found a cardinality constraint.
- Cardinality constraints are the ones in the form:

$$p_1 + \ldots + p_n \overset{\leq}{\underset{\geq}{=}} k$$

CARDINALITY CONSTRAINTS

- In sat these constraints are represented by:
  - at_least_one, at_most_one and $exactly\_one$;
  - at_least_k, at_most_k and $exactly\_k$.
- The encodings we presented can be very inefficient with big instances, so we need to find better ones.

CARDINALITY CONSTRAINTS

- Keeping in mind that:

$$at\_least\_k([x_1, \ldots, x_n], k) \equiv at\_most\_k(\{\neg x_i \mid x_i \in [x_1, \ldots, x_n]\}, n - k)$$

$$exactly\_k([x_1, \ldots, x_n]) \equiv at\_most\_k([x_1, \ldots, x_n]) \wedge at\_least\_k([x_1, \ldots, x_n])$$

- We are going to focus just on the at_most_k constraint.

AT MOST ONE

AT MOST ONE-PAIRWISE ENCODING

- The pairwise(or naive) encoding of the at_most_one constraint is:

$$\bigwedge_{1 \le i < n} \bigwedge_{i+1 \le j \le n} \neg(x_i \wedge x_j)$$

- This encoding doesn't require the addition of any new variables, but it encodes $O(n^2)$ clauses.

AT MOST ONE-SEQUENTIAL ENCODING

- The sequential encoding of the at_most_one constraint consists of using $n-1$ variables s_i to keep track of which x_i is true, it is encoded as follows:

$$(\neg x_1 \vee s_1) \wedge (\neg x_n \vee \neg s_{n-1}) \wedge \bigwedge_{1<i<n} ((\neg x_i \vee s_i) \wedge (\neg s_{i-1} \vee s_i) \wedge (\neg x_i \vee \neg s_{i-1}))$$

- This encoding produces $3n-4(O(n))$ clauses.

AT MOST ONE-BITWISE ENCODING

- The bitwise encoding of the at_most_one constraint consists of using $m = log_2(n)$ new variables $r_1, \ldots, r_m$ to represent the binary encoding of the index of the variable which is true, so it is encoded like:

$$\bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq m} \neg x_i \vee r_{i,j}[\neg r_{i,j}]$$

- Where $r_{i,j}[\neg r_{i,j}]$ if bit j of the binary encoding of $i - 1$ is 1[0].
- This encoding produces $nlog_2(n)$ clauses.

AT MOST ONE–HEULE ENCODING

- The Heule encoding is another linear version of the at_most_one constraint applicable for $n > 4$, which consists of splitting the pairwise encoding in two parts, adding an auxiliary variable y and repeating recursively the method for the second term, until the condition $n \leq 4$ is met:

$$at\_most\_one(x_1, \ldots, x_3, y) \wedge at\_most\_one(\neg y, x_4, \ldots, x_n)$$

- This encoding require the addition of $(n-3)/2$ new variable, but it encodes $3n-6$, $O(n)$ clauses.

AT MOST K

AT MOST K-PAIRWISE ENCODING

- The pairwise(or naive) encoding of the at_most_k constraint is:

$$at\_most\_k([x_1, \ldots, x_n], k) \equiv \bigwedge_{M \subseteq \{1,\ldots,n\}} \bigvee_{i \in M} \neg x_i$$

- This encoding doesn't require the addition of any new variables, but it encodes $\binom{n}{k+1}$ clauses of length $k+1$, with $|M| = k+1$.
- $O(n^{k+1})$ clauses.

AT MOST K-SEQUENTIAL ENCODING

- The sequential encoding of the at_most_k constraint consists of using $(n-1)*k$ variables $s_{i,j}$ to keep track of which x_i is true, and what sum j is reached at index i. it is encoded as follows:

$$
\begin{array}{ll}
(\neg x_1 \vee s_{1,1}) & \\
(\neg s_{1,j}) \quad \text{for } 1 < j \leq k & \\
\left.\begin{array}{l}
(\neg x_i \vee s_{i,1}) \\
(\neg s_{i-1,1} \vee s_{i,1}) \\
\left.\begin{array}{l}
(\neg x_i \vee \neg s_{i-1,j-1} \vee s_{i,j}) \\
(\neg s_{i-1,j} \vee s_{i,j})
\end{array}\right\} \text{for } 1 < j \leq k \\
(\neg x_i \vee \neg s_{i-1,k})
\end{array}\right\} & \text{for } 1 < i < n \\
(\neg x_n \vee \neg s_{n-1,k}) &
\end{array}
$$

- This encoding needs $2nk + n - 3k - 1$ clauses.